

FAST MODE DECISION FOR MULTIVIEW VIDEO CODING BASED ON SCENE GEOMETRY

Gianluca Cernigliaro, Fernando Jaureguizar, Julián Cabrera, Narciso García

Grupo de Tratamiento de Imágenes, Universidad Politécnica de Madrid, Madrid, Spain
{glc, fjn, jcq, narciso}@gti.ssr.upm.es

ABSTRACT

A new fast mode decision (FMD) algorithm for multi-view video coding (MVC) is presented. The codification of the views is based on the analysis of the homogeneity of the depth map and corrected with the motion analysis of a reference view, which is encoded based on traditional methods and on the use of the disparity differences between the views. This approach reduces the burden of the rate-distortion motion analysis using the availability of a depth map and the presence of the disparity vectors, which are assumed to be provided by the acquisition process.

Index Terms— Multiview Video Coding, H.264, Fast Mode Decision, Depth Map, Disparity Vector

1. INTRODUCTION

Multi-view video coding (MVC)[1] has received a renewed interest in the research and in the standardization communities as a consequence of emerging new paradigms for visual communications, including free viewpoint TV, 3D video, etc. In this paper, we address a scenario where additional data related to the 3D scene, namely disparity fields or depth maps, have to be computed to fulfill other application requirements apart from those imposed by the MVC encoding. This can be the case of free viewpoint video/TV applications where these data are needed for rendering purposes. Our objective then is to take advantage of the availability of such additional information to reduce the computational burden required by the MVC encoding.

The goal of a Fast Mode Decision (FMD) algorithm is to reduce the complexity of the Mode Decision (MD) stage to make faster the encoding process. Numerous FMD techniques are referenced in the literature which use early stop criteria [2], rate-distortion (RD) computations [2][3] or particular prediction algorithms [4]. With the new possibilities that a multi-camera scenario provides, many studies focused on the inter-view redundancies, reducing the heaviness of the process using, for example, the disparity vectors (DV's) [5][6] or the depth field [7].

The proposed algorithm targets MVC environments in which depth and disparity information are available, so, applying it on a couple of views, it is possible to achieve lower computational complexity in the MD stage.

This work has been partially supported by SAPEC and the Spanish Administration agency CDTI under project CENIT-VISION 2007-1007, and by the Ministerio de Ciencia e Innovación of the Spanish Government under project TEC2007-67764 (SmartVision).

2. DISPARITY AND DEPTH BASED FAST MODE DECISION (DDFMD) ALGORITHM

An MVC environment provides additional information that, usually, is not available in a normal mono-view coding scenario. In this paper a method is proposed which, taking advantage of such information, improves the performance of the coding process. Researches focused on the design of FMD algorithms oriented to the use of the depth field, which achieved good results in terms of quality and of computational charge. In the studies of Cheung *et al*[8], for example, using an MD based on the depth values, the mode selection gives a performance which, in terms of PSNR, is comparable to that obtained with a traditional MD algorithm. Also in 3D computer generated video has been fulfilled searches of depth based FMD techniques [9] able to reduce the complexity of the coding process without loss of quality. So, since there is correlation between motion field and depth field, working in an MVC scenario, in which such information is available and free, new techniques could be created to save up coding time preserving the PSNR.

In digital video coding, the MD step divides frames according to the motion differences of a sequence: areas where the motion is uniform are encoded with big block sizes and areas where it is not uniform are encoded with blocks of smaller dimensions. The goal of the proposed algorithm is to speed up the MD stage, analyzing the depth maps and taking into account that they are correlated to the motion fields. A depth map is a grey scale description of the scene using dark and bright values to represent, respectively, far and close objects. A non small difference between two depth values should represent a borderline between two objects. As, normally, areas of a frame which have uniform motion, probably, belong to the same object, it is possible to consider that a boundary marked by two depth values, corresponds to a boundary between two regions with different motion. In our [7] previous related work a Depth based FMD (DFMD) is proposed. In that case the mode selection takes into account the homogeneity of the depth, dividing into smaller blocks where such uniformity is not respected. In that algorithm the analysis is supported by another information which is the knowledge of the coding parameters of one of the other views of the scene, captured from another camera and slightly displaced with respect to the view which is actually in the coding process. The use of such support is due to the not always reliability of the depth maps which, sometimes, could represent areas with uniform motion with various depth values or, viceversa, dynamic areas with uniform depth values. So, in the DFMD, we assume that one of the views, called Reference View (RV), is encoded in a regular AVC/MVC fashion, where a motion based algorithm to evaluate the MD, which, after, is extrapolated and used to drive the depth based MD of the adjacent view, called Current View (CV). Now, in this paper, a new version of the Depth-based Fast Mode Decision (DFMD) [7] is proposed,

where the RV's MD and the depth maps are used together with the Disparity Vectors (DV) representing the spatial differences between two frames captured by two different cameras at the same time. The DV's are used to improve the use of the RV, which, in case of high disparity, could drive the MD to wrong partitions.

The algorithm description is structured as follows. In section 2.1, the partition step is shown, with the selection of the dimension of the blocks. Section (2.2) illustrates the decision of a threshold, evaluated through the RV's MD and the DV's to drive the MD. There are particular cases in which the DDFMD cannot be evaluated. In these cases the RV is used. Section 2.3 shows how to solve those critical situations.

2.1. Mode Decision based on the depth map

In this section we focus on the description of the partition algorithm for the MD, provided that a threshold has been previously set. The computation of the threshold will be described in the next section.

The size of the partitions depends on the motion variations of the scene. Our algorithm uses the correlation between the depth map and the motion. The idea is to divide every frame taking into account the analysis of its depth map. As has been shown in the previous work [7], the process starts by dividing the depth map into MB's and a splitting mechanism is applied on every one of them. In the first stage the homogeneity of the MB depth is tested, dividing it into 4 subblocks of 8×8 pixels. Their average values are computed and compared to the mean value of the MB. If the 4 differences are smaller than the threshold the MB is classified as a uniform motion region and encoded with the 16×16 mode, otherwise, is divided into four 8×8 partitions. The second stage is the comparison between vertical and horizontal couples of 8×8 subblocks, evaluating their differences and comparing them to the threshold. As the first stage, if the differences are lower than the threshold, the couples are classified as homogenous areas and, if both couples fulfill this condition the 8×16 or 16×8 mode is selected, depending on the placement of the pairs. If any partition, between the three shown previously, is not selected, the 8×8 mode is chosen and the smaller partitions of H.264 need to be tested. Therefore, the same procedure is applied, keeping the same threshold, on every 8×8 block to test the presence of the 4×8 , 8×4 and 4×4 partitions.

2.2. Threshold computation

As explained previously, the splitting process is not static. Due to the not always reliable characteristics of the depth, there is not the absolute certainty that every depth gap corresponds to a motion difference. Therefore, an adaptive mechanism has been created which changes the sensibility of the algorithm to the depth differences according to the movements of the scene. The goal of this mechanism is to evaluate, using the RV and the DV's, a threshold which should define a limit between the depth differences which correspond to a motion difference and the depth differences which belong to static areas.

This algorithm is based on the DFMD [7], which uses the same splitting technique to evaluate the MD, but, using a simpler threshold decision step that only takes advantage of the MD of the RV. In this section will be explained why it has been decided to add a mechanism based on the disparity and how it makes better the coding efficiency.

In a normal H.264 scenario, the MD process divides the frames into partitions according to the motion variations. When coding the RV using the traditional H.264 standard rules, a division evaluated considering only such characteristic of the sequence is available. Applying such division onto the depth map of the CV, it is possible to segment the depth regions according to the movement.

The goal of the threshold evaluation is to obtain a minimum depth difference which corresponds to a motion difference. Every depth MB by is divided into subblocks with different motions according to the MD of the RV; the next step is the evaluation of the maximum value for every subblock and, after the computation of the absolute differences between them, the minimum is selected as a threshold (1).

$$th_k = \min |d_i - d_j| \quad i, j = 1, 2, \dots, \#subblocks \quad (1)$$

where th_k is the threshold for the k -th MB and d_i and d_j are the maximum depth values of subblocks i, j within the k -th MB. Figure 1 shows the division of the depth MB and, after, how on this segmentation the maximum depth values are evaluated to select a threshold.

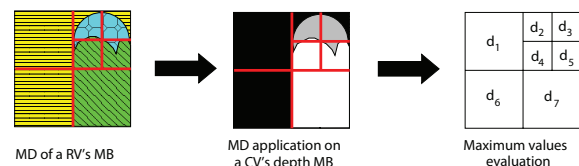


Fig. 1. Segmentation of the depth maps according to the RV's MD.

A threshold estimated using the described method provides good results if there is not disparity between RV and CV. But, if the objects in the two views are slightly displaced, the application of the MD of the RV on the depth map of the CV could provide a bad threshold due to the bad overlapping (Figure 2)

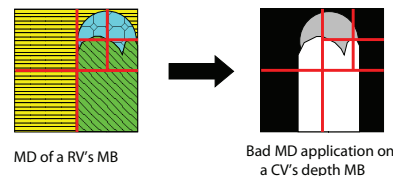


Fig. 2. Bad correspondence between RV and CV in case of disparity.

This problem is solved using the DV's. Such vectors correspond to the differences of the object's positions between the RV and the CV. The DV, for each MB in the CV, is used to find the corresponding area in the RV. The segmentation of the found region is overlapped on the corresponding MB of the depth map and the same threshold evaluation is applied. As shown in Figure 3, in this case there's not any error due to the disparity.

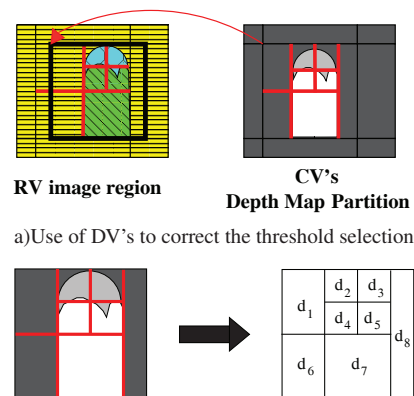


Fig. 3. Depth map's segmentation using disparity.

2.3. Analysis of the surrounding area

There are particular cases where it is not possible to evaluate the threshold. As explained in the previous section, the threshold is computed dividing the depth map and evaluating the differences between the maximum depth values. If the depth region is totally homogeneous or if there is not partition on the RV's MB it is impossible to evaluate differences and, as consequence, there are particular cases which is impossible to detect. For example, areas with different motion but same depth, or areas with depth gaps on which is applied a 16×16 partition to evaluate the threshold. In these cases the algorithm could apply a 16×16 mode supposing that there is not motion in the area. However, to improve the efficiency, in terms of quality, of the proposed algorithm, a correction mechanism is needed.

Such mechanism has been used in the DFMD [7]. So, to corroborate if the selection of a 16×16 mode is appropriate, as the RV is very similar to the CV, the mode used to encode the MB's surrounding the RV's MB is checked. If the the majority of them are coded with a 16×16 mode, then the probability to find a static region (or with homogenous motion) is very high, so the 16×16 mode is selected, else an MD algorithm implemented in the H.264 Reference Software [10] is applied.

3. EXPERIMENTAL RESULTS

The proposed FMD strategy has been implemented within the H.264/AVC Reference Software (RS) version 14.2 [10]. The configuration of the RS has been set using the Main Profile. The rate control has not been activated (fixed QP) and the allowed modes were only the Inter frame partitions (16×16 , 16×8 , 8×16 , 8×8 , 4×8 , 8×4 , 4×4) except the P-Skip Mode. All the simulations were carried out using a GOP of 90 Frames with one I-Frame and 89 P-Frames. The coding process takes into account the depth maps of the sequence and the DV's evaluated between the CV and the RV. DV's were MB-wise computed using a Motion Estimation like search algorithm between two different views.

Several MD algorithms are implemented in the RS which provide different results in terms of quality and coding time. The goal of the DDFMD is to reach the performance of a FMD algorithm speeding up the process. For this reason the comparison have been made considering the Fast High Complexity (FHC) MD [2], which has high quality performance thanks to the RD analysis but this implies a heavy computational burden. On the other hand, being the DDFMD an improvement of the DFMD [7], a comparison between the two versions of the technique is done. In the tables and in the graphics are shown respectively, the percentage of coding time saved evaluated as in Eq. 2, and the performance of the algorithm in terms of PSNR (dB).

$$\Delta t(\%) = \frac{t_{FHC} - t_{DDFMD}}{t_{FHC}} 100 \quad (2)$$

The sequences considered in the experiments, all designed for a MVC scenario, were Akko&Kayo, Pantomime and Breakdancers.

Table 1 shows the percentages of saved time by the proposed algorithm and by its older version respect to the FHC MD. In the coding process, when the DDFMD is used, it is not required to evaluate the RD to check all the allowed modes, so the time saved is considerable and, as it is possible to deduce by the results, the use of the disparity adds computational burden almost irrelevant with respect to the version which doesn't use such information.

The performance in terms of PSNR has been evaluated using different couple of sequences with different disparity situations. In the first experiment (Akko&Kayo), views number 2 and number 3 have been used as RV and CV respectively. In this case, the mean DV (evaluating the average of all the DV's between the two views)

QP	25	27	30	32	35	40	45	50
Akko&Kayo								
% Δt DFMD	33	33	34	34	35	37	38	40
% Δt DDFMD	33	33	34	35	35	37	38	40
Pantomime								
% Δt DFMD	20	21	23	23	24	25	27	27
% Δt DDFMD	15	17	20	20	21	23	25	26
Breakdancers, CV=3								
% Δt DFMD	14	14	15	17	19	22	21	21
% Δt DDFMD	4	6	10	13	16	20	19	19
Breakdancers, CV=7								
% Δt DFMD	15	14	16	18	20	22	22	22
% Δt DDFMD	4	7	11	13	16	20	19	19

Table 1. Computational time differences (%) between the DFMD and the FHC MD and between the DDFMD and the FHC MD.

is 14.86 pixels. As shown in Figure 4 the DDFMD algorithm obtains an improvement with respect to the DFMD.

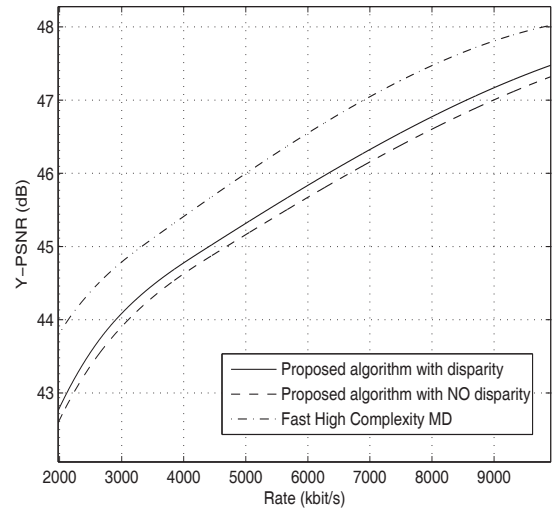


Fig. 4. Comparison among FHC, DDFMD and DFMD for the sequence Akko&Kayo.

Using couples of sequences with different mean DV's the performances could vary because, depending on the characteristics of the geometrical displacements and how the algorithm can correct them, the DDFMD improves the DFMD to a greater or lesser extent. For this reason the second case of analysis has been made on the sequence Pantomime, using views number 29 and number 30 as RV and CV (average DV=18.4 pixels). Figure 5 shows the performance of the new method which, in this case, reaches and, for the high bit rates, improves the FHC MD graphic. The presence of a higher displacement between the two cameras is well managed by the disparity correction which, also in this case, improves the coding performance of the DFMD.

The last experiment has been carried out on two couple of views of the same sequence (Breakdancers), using the same RV in both cases. The DDFMD has been applied, first, on the couple RV=2 and CV=3 (average DV=16.75 pixels), and, then, on the couple RV=2 and CV=7 (average DV=29.15 pixels). This last analysis has been made to examine the performance of the DDFMD, with respect to the DFMD, in two situations in which the video coding features are similar, but the geometrical differences change because the videos

are captured by different couples of points of views. Figure 6 shows the comparison of both cases. The results of the simulation follow the same trend of the previous comparisons: the disparity correction of the DDFMD is able to improve the DFMD in different displacement situations.

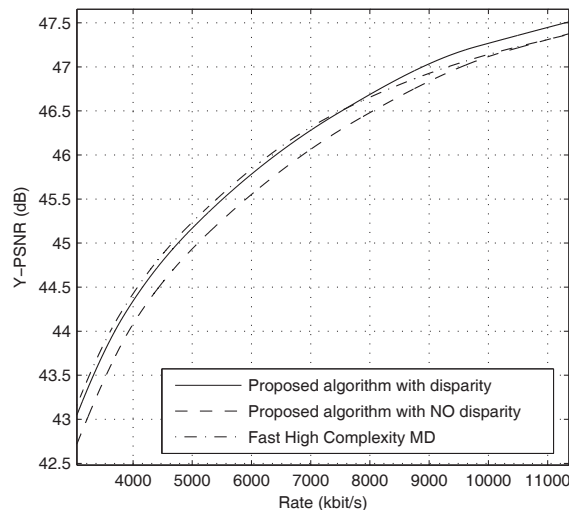


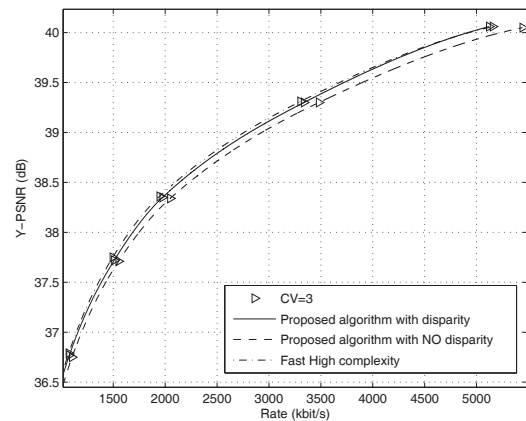
Fig. 5. Comparison among FHC, DDFMD and DFMD for the sequence Pantomime.

4. CONCLUSIONS

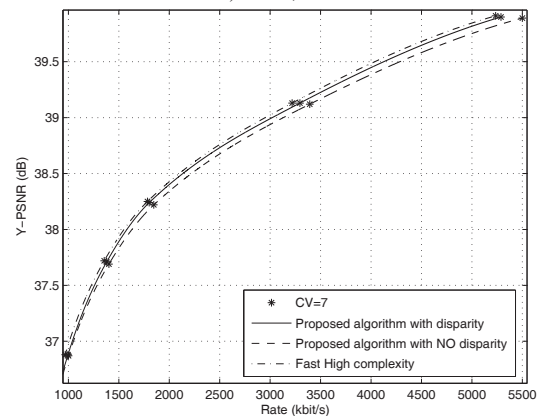
A new FMD strategy is proposed. Thanks to the characteristics of a MVC scenario, the proposed method is able to save time with respect to the FHC MD implemented in the RS. The technique uses information of the depth maps, and of the disparity vectors evaluated between the two views involved. Using the additional information of a reference view, coded in a normal H.264 fashion, a significant quantity of computational time has been saved. Experiments show that the DDFMD, applied on one view of a MVC scenario, is able to speed up the coding process obtaining variable results in terms of quality, depending on the sequence. In the majority of the cases, DDFMD PSNR is lower than the obtained with the FHC MD, but, in some cases the DDFMD reaches such performance and, in others, improves it. Has also been made a comparison on two couple of views of the same sequence, to analyze different geometry situations. Experiments show as, different displacement do not affect the new algorithm which enhance the PSNR in both cases.

5. REFERENCES

- [1] A. Vetro, P. Pandit, H. Kimata, A. Smolic, and Y. Wang, "Joint draft 7.0 on multiview coding," *Doc. JVT-AA209, Geneva, Switzerland*, April 2008.
- [2] Peng Yin, H.-Y.C. Tourapis, A.M. Tourapis, and J. Boyce, "Fast mode decision and motion estimation for jvt/h.264," in *Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on*, Sept. 2003, vol. 3, pp. III-853-6 vol.2.
- [3] K. P. Lim, G. Sullivan, and T. Wiegand, "Text description of joint model reference encoding methods and decoding concealment methods, jvt-o079," Tech. Rep., April 2005.
- [4] M. Nieto, L. Salgado, and J. Cabrera, "Fast mode decision on h.264/avc main profile encoding based on psnr predictions," in *Image Processing, 2006 IEEE International Conference on*, Oct. 2006, pp. 49-52.



a) RV=2, CV=3



b) RV=2, CV=7

Fig. 6. Comparison among FHC, DDFMD and DFMD for the sequence Breakdancers.

- [5] Li-Fu Ding, Pei-Kuei Tsung, Shao-Yi Chien, Wei-Yin Chen, and Liang-Gee Chen, "Computation-free motion estimation with inter-view mode decision for multiview video coding," in *3DTV Conference, 2007*, May 2007, pp. 1-4.
- [6] Li-Fu Ding, Pei-Kuei Tsung, Shao-Yi Chien, Wei-Yin Chen, and Liang-Gee Chen, "Content-aware prediction algorithm with inter-view mode decision for multiview video coding," *Multimedia, IEEE Transactions on*, vol. 10, no. 8, pp. 1553-1564, Dec. 2008.
- [7] Gianluca Cernigliaro, Fernando Jaureguizar, Antonio Ortega, Julian Cabrera, and Narciso Garcia, "Fast mode decision for multiview video coding based on depth maps," 2009, vol. 7257, p. 72570N, SPIE.
- [8] G. Cheung, T. Sakamoto, and W. Tan, "Graphics-to-video encoding for 3g mobile game viewer multicast using depth values," in *Image Processing, 2004. ICIP '04. 2004 International Conference on*, Oct. 2004, vol. 4, pp. 2805-2808 Vol. 4.
- [9] G. Cheung, A. Ortega, and T. Sakamoto, "Fast H.264 mode selection using depth information for distributed game viewing," in *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, Jan. 2008, vol. 6822 of *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*.
- [10] Fraunhofer-Institute HHI, "H.264/avc jm reference software (jm 14.2)," 2009.